On a Taxonomy of Delegation

Quan Pham, Jason Reid, Adrian McCullagh and Ed Dawson

Abstract Delegation, from a technical point of view, is widely considered as a potential approach in addressing the problem of providing dynamic access control decisions in activities with a high level of collaboration, either within a single security domain or across multiple security domains. Although delegation continues to attract significant attention from the research community, presently, there is no published work that presents a taxonomy of delegation concepts and models. This paper intends to address this gap.

1 Introduction

Traditionally, delegation may be used as a term for describing how duties and the required authority propagate through an organisation. In technical settings, often the term *delegation* is used to describe how an entity passes some specific capabilities on to another entity. However, delegation in technical settings is an ill-defined concept. Currently, there is no single study that provides a comprehensive taxonomy of delegation concepts and models. Thus, there is a need for a taxonomy which acts

Quan Pham
Information Security Institute, Queensland University of Technology, Brisbane, QLD, Australia, e-mail: q.pham@isi.qut.edu.au

Jason Reid
Information Security Institute, Queensland University of Technology, Brisbane, QLD, Australia, e-mail: jf.reid@isi.qut.edu.au

Adrian McCullagh
Information Security Institute, Queensland University of Technology, Brisbane, QLD, Australia, e-mail: a.mccullagh@isi.qut.edu.au

Ed Dawson
Information Security Institute, Queensland University of Technology, Brisbane, QLD, Australia, e-mail: e.dawson@isi.qut.edu.au

as a conceptual framework to help researchers position their research. This paper proposes a set of taxonomic criteria which can then be used to analyse a range of delegation proposals and models. This paper also investigates a number of delegation approaches from various perspectives such as actors, credentials, attributes, protocols, etc. to characterise each approach.

In this paper, for purposes of precision and clarity, we adopt the terminology used in the XACML specification. *Attributes* will be used to describe the following information: group, role and other information which can be ascribed to a particular entity. The entity that performs a delegation is referred to as a *delegator* and the entity that receives a delegation is referred to as a *delegatee*. An attribute will be said to be *delegatable* if it can be successfully granted from one entity to another.

The rest of the paper is organised as follows. Section 2 presents a taxonomy for delegation support in information systems. Section 3 discusses some notable works and their characteristics in the field and maps them with the characteristics described in the taxonomy. Section 4 discusses notable characteristics of these approaches and future trends in development of the delegation concept. Section 5 concludes the paper.

2 A Taxonomy

This paper is concerned with the implementation of the delegation concept in technical settings. This paper considers delegation as *a proxy process in which one entity grants/allocates the necessary attributes to another entity to enable the receiver to be able to perform certain responsibilities or capabilities while meeting certain obligations and constraints (e.g. with respect to duration, frequency etc.).* A delegation process usually includes a mechanism to revoke the delegated attributes (revocation). This section discusses in detail each dimension of the taxonomy which are summarised in Table 1.

Motivation Depending on the type of the operational environment, there may be different factors motivating delegation between the entities. From a technical point of view, these include:

- *Lack of authorisation* An entity does not have sufficient authorisation to perform certain actions over certain resources to complete a task.
- *Lack of or conflicted policies* Policies required to achieve a certain goal may conflict and the entity involved with the activities may need to delegate the tasks to another entity which is not affected by the conflicted policies.

Delegation Boundary The delegation can happen *within a single security domain*, or *across multiple security domains*. Delegation within a single security domain is the simplest case and is relatively easy to manage because of the centralised storage of policies and credentials. Until recently, most proposals restrict their scope to delegation within a single security domain. As the issues of security for collaborative environments have emerged, the concept of delegation needs to be considered

Table 1 Characteristics of Delegation

Characteristic	Factor
Motivation	- Lack of Authorisation - Lack of or Conflicted Policies
Delegation Boundary	- Within a security domain - Across multiple security domains
Who requests delegation?	- User - System Authority
Who delegates?	- User (Ad-hoc) - System Authority (Administrative)
Relationship of the parties	- Direct - Indirect
What to delegate?	- Capability - Responsibility
How much to delegate?	- Partial - Total
How long to delegate?	- Temporal - Permanent
How to delegate?	- Transfer - Grant
Authority Pre-Approval	- Yes - No (Optimistic Delegation)
Type of Credential	- X.509 - SAML Assertion - Generic Token
Key Scheme	- Symmetric Key - Asymmetric Key
Where delegation happens?	- User Level (Application) - System Level
Where is delegation honoured?	- Access Decision Point - An additional authority for delegation

from a new angle: delegation across multiple security domains, e.g. an entity from one system can delegate to another entity on another system. Cross domain delegation can bring flexibility to collaborative activities and can meet the needs of such dynamic environments [10]. However, cross domain delegation must cope with the complexity in building delegation protocols and exchanging/validating delegation tokens due to the potential inconsistency of security approaches by different systems.

Who requests delegation? As the motivations discussed above can happen with both normal users and the system authority, delegation can be requested by either a *user* or a *system authority*. Delegation requested by users is common. Consider for example, when a CEO employs a company secretary, he will want to allocate certain duties to the company secretary, for example, preparing the annual financial report. In a sense, this is the allocation of responsibility (a granting process) from the CEO to the company secretary. In contrast, delegation from the system authority is considered as a special case. The delegation of system authority is fixed and to

some extent, well pre-defined by the organisation's policies and procedures. In this type of delegation, the system authority actually does not request the delegation for itself; in fact, it requests the delegation on behalf of the delegator and the delegatee.

Who delegates? (Who is the delegator?) The delegator can be a *user (ad-hoc)*, or *system authority (administrative)*. Schaad argued that user ad-hoc delegation and administrative delegation can be differentiated based on three factors [11]: the representation of the authority to delegate, the specific relation of the delegator to the delegated attributes, and the duration of the delegation (how long the delegated attributes can last?)

Administrative delegation is the basic form of delegation in which an administrator or system authority assigns attributes and privileges to enable users to conduct certain tasks. This process typically happens when a user joins a security domain. The delegator, in this case, represents the authority of the system (system administration). In user delegation, the delegator is a normal user. So the delegator represents the authority of the user only. This is the case in which a user grants or transfers the whole or a subset of his/her attributes to other users. As the user is the delegator, the user must possess the ability to utilise the attributes to be able to perform delegation. This type of delegation is typically short-lived and intended for a specific purpose [6, 11].

Relationship of the parties in the delegation process The relationship between the delegator and the delegatee can be considered as either *direct delegation* or *indirect delegation (sub-delegation)*. Direct delegation is defined as the delegation in which the delegator directly sends the delegation assertion to the delegatee. In contrast, indirect delegation is performed with the involvement of one or many intermediate parties which can forward the delegation assertion from the delegator to the delegatee. This type of delegation is sometimes called sub-delegation. Indirect delegation is mainly performed to achieve a *multi-step delegation*. Indirect delegation is especially important in the context of cross domain delegation when the delegation token traverses various security domains.

What to delegate? What to delegate is the main and the most controversial topic in the field. The object of the delegation process is a key aspect on which proposed models differ. The following three main cases are evident in published proposals:

- *Case 1*: The delegatee takes on attributes of another entity (the delegator) via an unforgeable token which has the capability to perform the task.
- *Case 2*: The delegatee is assigned some attributes that the authority will evaluate in the context of a set of applicable policies. The difference to Case 1 is that the delegated attributes are considered as new attributes of the delegatee while in Case 1, the delegated attributes are treated as if they are from the delegator.
- *Case 3*: The delegatee is assigned new responsibilities as part of the delegation commitment between the involved parties or part of the constraints set by the applicable policies. It is very often that the attribute that represents new responsibility is "role". This case, however, more precisely reflects the social nature of the delegation concept.

To stimulate the above cases, at the abstract level, there are two trends:

- *Delegation of Capability*: Case 1 and Case 2 represent a type of delegation of capability as the delegation will enable new capability in the delegatee. In this paper, the term *capability* is used in the sense which it is defined in POSIX Draft 1003.1e/2c as simply a representation of the ability to perform a specified task.
- *Delegation of Responsibility (Case 3)*: It is a form of transferring tasks as well as obligations/conditions or commitments which are associated and covered by certain responsibilities from one entity to another [1, 7].

In general, delegation of capability is technically well defined. This type of delegation is defined to cope with the demand for a high level of granularity and is appropriate for environments which require a high level of flexibility. However, delegation of responsibility is considered as a broader concept compared to delegation of capability. From the responsibility perspective, the process is defined via the responsibility to transfer or grant and it is assumed that necessary attributes or rules to complete the duties will be transferred or granted upon completing the process. The associated obligations/conditions or commitments are considered as part of the delegation process.

How much to delegate? In general, depending on the needs of the delegator and delegatee, the delegation can be *partial* or *total delegation*. Partial delegation can be achieved by delegating just a specific subset of capabilities/responsibilities. On the other hand, total delegation can be achieved by delegating the whole set of capabilities/responsibilities associated with certain attributes. Total delegation is the extreme case. In fact, the concepts of partial and total delegation are quite relative.

How long to delegate? Delegation can be *temporal* or *permanent*. Temporal delegation is a time-constrained delegation of which the validity period is set by either the delegator or the system authority. On the other hand, permanent delegation is a type of delegation which does not need a specified expiry time. The delegation and revocation process is triggered by a specified event. This type of delegation can be considered as relatively permanent. Permanent delegation is usually associated with administrative delegation due to the nature of the relationship of the delegator and the delegatee and the organisation's policies. In ad-hoc delegation, permanent delegation is rare and is usually considered as a failure of the system to reflect a change in circumstances.

How to delegate? From the operational perspective, delegation may be classified into two categories: *grant delegation* or *transfer delegation* [6]. In grant delegation, a successful delegation operation allows a delegated attribute to be available to both the delegator and delegatee. So after a grant delegation, both delegatee and delegator will share some common attributes. Grant delegation makes the availability of attributes increase monotonically with delegations [6]. Grant delegation is primarily concerned with allowing the delegatee to use the delegated attributes. On the other hand, in transfer delegation, besides allowing the delegatee to use the delegated attributes, the mechanism must be able to prevent the use of the delegated attributes by the delegator.

Authority Pre-Approval Delegation can be *pre-approved* or *optimistic*. At the time a delegator receives a delegation request, it does not necessarily know in advance whether a particular set of delegated attributes will be useable by the delegatee, since it may not have a complete understanding of the current security context of the delegatee, the current set of attributes of the delegatee, and the policies of the delegatee's systems, etc. To avoid making a delegation that will not be honoured, the delegator could contact the relevant authorisation authorities to ask "if I delegate these attributes to user X from domain Y, will they be honoured?" But asking this question in advance for each delegation transaction is inefficient as the authorisation authority will then need to evaluate the request twice - once for the pre-approval and once for the actual execution by the delegatee. Therefore, in optimistic delegation, the delegator agrees to conduct the delegation transaction on the basis of its best knowledge of the constraints and conditions for the delegation transaction, for example, the policies of its systems, the attributes, etc. It does not guarantee that the delegatee will be able to successfully use this attribute for service invocation.

Type of Credential In general, there are three forms of credentials which are commonly used to bear delegation information: *X.509*, *SAML assertion*, and *generic token*. The generic token is a signed statement that includes the public keys of the delegator and the delegatee, the involved attributes and a timestamp. Over time, the delegation credential has become more sophisticated. Currently, most proposals use the SAML assertion and more popularly X.509-based attribute certificate (such as in the PRIMA [9] and PERMIS [4] systems) as the means to bear the delegation credential. It is worthy to note that the generic form of delegation token above can be only useful in a single delegation transaction. For a multi-step delegation with the involvement of multiple intermediate entities, it is essential to employ a more complex form of delegation token via a different combination of multiple delegation tokens.

Key Scheme In general, keys play an important role in securing the exchanged delegated attributes between the delegator and the delegatee. Keys are primarily used to encrypt and sign the delegation tokens. Currently, due to the increasingly popular and well standardised PKI with X.509 certificate, asymmetric key scheme seems to be the default option for constructing delegation protocols. However, Varadharajan suggests that both *symmetric key* and *asymmetric key* schemes can be used either separately or in combination in a hybrid form to support the delegation process [12, 13]. The symmetric key approach is somewhat similar to the asymmetric key approach, in that the underlying principle of signing or encrypting the delegation token is the same. However, in this case, the secret key used to encrypt or sign the delegation token is assumed to be shared between the delegator and the delegatee and issued by a trusted third party which can be the system authority.

Where delegation happens? The delegation can happen at multiple levels: *system level* and *user/application level*. At the system level, the delegation is classic in the sense that the delegation is pre-defined in a concise manner. This type of delegation is often limited to a set of well studied scenarios. In the system level, delegation

usually happens as part of the supported access control model, for example, adding a user to a group in UNIX. Delegation at this level is considered as part of the access control infrastructure but there is a lack of flexibility to cope with unconventional scenarios, especially in collaborating activities with external parties. This is where delegation at the user level can make a difference. Delegation at the user or application level is usually ad-hoc in nature and is necessary to address the flexibility of the access control system. At the user or application level, people may need to accommodate not only different technical standards but also different workflows, business processes and frameworks. In this context, delegation is an essential element in business processes which require a high level of collaboration. In general, workflows control the execution of business processes in an organisation at the technical or information system level [1, 3].

Where is delegation honoured? In general, any access control system is centred around the following two functions: access decision function and access enforcing function. Commonly, they are also known as Policy Decision Point (PDP) and Policy Enforcement Point (PEP) respectively. Therefore, when a request is associated with a delegation, the validation process can be conducted at: *Policy Decision Point (PDP)* with the partial contribution of the PEP or *an additional authority* which governs for delegation transactions.

In theory, it is safe to consider the PEP as part of the delegation validation process. This is because the PEP is the authority who receives the request from the user (the requestor). From this point of view, the PEP is the one who is responsible for receiving the credentials from users and passing them to the PDP for decision making. On the other hand, the PDP is responsible for evaluating the policy (also taking into account the credentials provided by users/subject). Most delegation-supporting access control models consider the validation process as an additional function of the PDP.

The second approach is to use an additional authority such as the Credential Validation Service [5] or the Delegation Authority [8] to govern the delegation function. For example, the Credential Validation Service could be incorporated into the XACML model. In fact, the PDP is still responsible for decision making. However, in this approach, the PEP is not the authority to collect and transfer the delegation credential to the PDP for decision making. This role now belongs to the new delegation authority. In the context of XACML, the delegation authority could also act as a replacement of the Policy Information Point (PIP). The advantage of this approach is that systems with existing access control models do not need to change. The only change is to provide an interface to call and respond to the delegation authority. In the design of Chadwick et al., the Credential Validation Service could be either an additional component to be called by the PEP or the PIP [5].

Table 2 Comparison of some notable delegation approaches using the taxonomy's characteristics

Characteristics	Varadharajan, Allen & Black's Model	PBDM Family	Atluri and Warner's Model	Gomi et al.'s Model	Chadwick's Model
Motivation	Lack of authorisation	Lack of authorisation	Lack of authorisation and conflicted policies	Lack of authorisation	Lack of authorisation and conflicted policies
Delegation Boundary	Within a single domain	Within a single domain	Within and cross security domains	Within and cross security domains	Within and cross security domains
Who requests delegation?	User	User or System authority	Mainly focus on User level	User	User
Who delegates?	User	User or System authority	User	User	User
Relationship of the parties	Both direct and indirect	Both direct and indirect	Both direct and indirect	Both direct and indirect	Direct. Indirect delegation is not clearly discussed.
What to delegate?	Capability or Responsibility	Capability and Responsibility	Capability and Responsibility	Capability and Responsibility	Capability and Responsibility
How much to delegate?	Partial or Total	Partial or Total	Partial or Total	Partial or Total	Partial delegation is not specified
How long to delegate?	Temporal or Permanent	Temporal or Permanent	Temporal or Permanent	Temporal or Permanent	Temporal or Permanent
How to delegate?	Grant	Grant	Grant or Transfer	Grant	Grant
Authority Pre-Approval	Yes	Not specified	Not specified	Yes	Partially discussed
Type of Credential	Generic Token	Not specified but can be any	Not specified but can be any	X.509, SAML or Generic token	X.509, SAML or Generic token
Key Scheme	Symmetric or Asymmetric	Not specified but can either	Not specified but can either	Symmetric or Asymmetric	Symmetric or Asymmetric
Where delegation happens?	User level	Both but mainly target the System level	User level	User level	User level
Where delegation is honoured?	Not specified	Not specified	A central authority based on RBAC	An additional component called *Delegation Authority*	An additional component called *Credential Validation Service*

3 Some Approaches to Delegation Problem and Classifications

In this section, the taxonomy criteria are utilised to compare some notable delegation approaches. For brevity, this paper does not discuss in detail each approach, but instead gives a brief discussion about the notable features and characteristics of each approach based on the taxonomy dimensions presented in Table 2.

Varadharajan, Allen and Black's work in 1991 discussed in detail how a protocol for delegation should be structured [13]. Based on the taxonomy, it can be said that the model of Varadharajan, Allen and Black is specifically designed to support *both key schemes*. From delegation perspective, the work, via the delegation of privilege, does not clearly explain the objective of the delegation process (*capabil-*

ity or responsibility). It also fails to explicitly discuss *the relationship of delegator and delegatee*. While the protocol has the potential to extend to cover *cross domain transactions*, it does not cover this issue in detail.

Zhang, Oh and Sandhu presented a new permission-based delegation model (PBDM) in 2003 [15]. This model fully supports user to *user, temporal, partial and multi-step delegation*. This model is later extended and presented in three variants called PBDM0, PBDM1 and PBDM2. The PDBM family can support *multi-step delegation*, but they neither support constraints in delegation, nor *delegation across multiple security domains* [6]. In this model, both types of *grant and transfer delegation* are supported. The PBDM family can be considered as an extension of the RBDM [2] and RDM [14] models.

In 2005, in an effort to address constraint issues in delegation, Atluri and Warner [1] studied delegation in the workflow context and introduced a conditional delegation model. This is an interesting delegation approach as it investigates the problem of delegation with an *ad-hoc nature*. This is also one of the first models that details how delegation should be handled at the *user level* and *how/where the delegation should be honoured*. This model is one of the pioneers in the field that address the issue of delegation in the workflow context. However, similar to previous models, this work also fails to discuss *the relationships* between the delegator, the delegatee and the service provider.

Gomi et al. presented a basic framework to conduct *grant delegation* and revocation of access privileges *across security domains*. The model of Gomi et al. [8] requires the delegator to request the delegation assertion via *an additional authority called Delegation Authority (DA)*. This model lacks the capability to check for constraints and resolve conflicts between delegated privileges and between the delegated privileges with the involved policies. Therefore, it can cause problems in *indirect delegation which happens across multiple security domains*. The issue of *authority pre-approval* in the delegation process is partially discussed via the appearance of the delegation authority.

As part of efforts to develop PERMIS, Chadwick et al. proposed a mechanism based on the XACML conceptual and data flow models to address the issue of dynamic delegation of authority which involves the issuing of credentials from one user to another (*user delegation*) [5]. They proposed a new conceptual entity called the *Credential Validation Service*, to work alongside the PDP in making authorisation decisions. The model does not support *indirect delegation* well. Similarly to Gomi et al.'s work, this model, via the Credential Validation Service, partially discusses the issue of *authority pre-approval* but does not explicitly describe how delegation can happen without the pre-approval.

4 Discussion

To date, most delegation models have been centralised and based on the RBAC model. Delegation of capabilities seems to be a major concern of most models, ex-

cept for some recent delegation models for workflow such as the works of Atluri and Warner, Gomi et al. and Chadwick et al. Most models have problems with partial and user (ad-hoc) delegation. It is also worth noting that, until 2004, most published works regarding delegation focused primarily on delegation between entities within a single security domain. More recently, there has been a trend toward increasing focus on cross domain issues. It can be seen that most recently developed models such as Atluri and Warner, Gomi et al., Chadwick et al., etc. are purposefully designed to support cross domain delegation.

Cross domain delegation is designed to achieve flexibility to meet the demand of collaborative activities. However, it is much more complicated to implement and enforce constraints over the ad-hoc delegation in cross domain models (Chadwick et al. vs. Varadharajan, Allen and Black). In addition to the same issues of classic delegation (within a single domain), the complexity of protocol and policy is a paramount issue. Such complexity requires a very well designed protocol and a high level of agreement between systems. To achieve cross domain delegation, the involved authority must also take into account the distribution of applicable policies across various security domains. For example, if the delegatee, the delegator and the service provider reside on three different security domains, all policy sets of these three domains must be considered and fed to the authority in charge of the delegation process for any decision making. This process is quite simple in single-domain delegation as there is only one single authority to handle the storage of credentials and feed them to the access decision authority. However, in addition to the distribution of policy, credential and delegation information are also distributed in cross domain delegation. The typical scenario is that the delegator and its local authority store and maintain part of the delegation information related to the delegator while the delegatee and the authority of the delegatee's domain store and maintain the rest. It is important to note that the main characteristics of delegation, such as delegation boundary, where delegation happens and where delegation is honoured, have a significant impact on making design decisions. This is because these factors are vital to form the backbone for a flexible and scalable cross domain delegation solution.

Together with the current trend in supporting cross domain collaborating activities, it is also important to note that there is an increasing demand in providing context-aware information to accommodate constraints and commitments for the delegation process. As current role-based approaches use the relationship of user-role-permission to impose constraints, it is difficult to present the additional context information to the access decision authority. Thus, there is a demand for a more expressive approach than the current role-based mechanisms. This is the reason why recent approaches such as Chadwick et al. (using XACML) or Gomi et al. (using SAML), etc. have adopted the policy language-based approach. With well defined languages such as XACML, SAML, etc., these models show that they can better address the issue of constraints. Even though policy language-based communication is exposed to high overhead and may result in low performance, this may be the only feasible approach to address the needs of highly collaborative activities across multiple security domains where constraints and context-awareness are critical. Depending on the level of application of a policy language-based approach, each model

achieves a different level of expressiveness. The positive effects of applying the policy language-based approach can be seen clearly in Chadwick et al.'s model against the classic role-based approach in PBDM or RBDM family. However, application of a policy language is not the sole factor that determines the usefulness of a model because there are other factors that affect the final outcomes such as how the language is implemented, to what extent the language is implemented, the power of the language itself, etc.

5 Conclusion

This paper discussed the concept of delegation via a number of dimensions and presented a taxonomy of delegation concepts in the context of information systems and applied it to several delegation proposals. The taxonomy can be used to understand the major focus of a particular delegation approach by observing the characteristics involved. The taxonomy can help raise awareness of various design settings and potential implications on existing access control infrastructures.

Therefore, it can be said that this study is significant for several reasons. First of all, with the emerging demands in federating multiple enterprise systems together to achieve complex and collaborative activities, delegation is becoming a common approach to provide dynamic and flexible access control decisions. Secondly, delegation is considered a comparatively new research area and requires more input from the academic and industrial community and, although recent research has addressed the problems, several issues still remain to be investigated and resolved. Therefore, this research should provide system designers a clear picture about the characteristics of different types of delegation approaches and the involved actors so that they can choose the type of delegation that best satisfies their requirements. Thirdly, as collaboration environments require a great level of interoperability, knowledge of characteristics and protocols of different types of delegation could vastly improve the integration process.

Finally, as the main focus of this paper is delegation approaches which can be used in secured task distribution in workflow or secure ad-hoc collaboration, currently, this paper does not cover the complete set of delegation approaches with the ad-hoc nature that can be applied highly dynamic and ad-hoc transaction such as secure task distribution in workflow or secure collaboration. Therefore, as the future work, some other aspects of delegation will be considered such as the rubric of trust management, logic-based and cryptographic approaches.

Acknowledgements The research is kindly funded by the Smart Services CRC, the Information Queensland, Queensland State Government, Australia and the Australian Research Council - Project *DP0773706*. This paper has been abridged due to space constraints. The full version of the paper can be found in the technical report at QUT ePrints - *http://eprints.qut.edu.au/17213/*.

References

1. Atluri, V. and Warner, J.: Supporting conditional delegation in secure workflow management systems: In *Proceedings of the 10th ACM symposium on Access control models and technologies (SACMAT'05)*, Stockholm, Sweden, 49–58 (2005).
2. Barka, E. and Sandhu, R.: Role-Based Delegation Model - Hierarchical Roles (RBDM1): In *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04)*, 396–404 (2004).
3. Botha, R. A. and Eloff, J. H. P.: A framework for access control in workflow systems. Information Management & Computer Security, **9(3)**, 126–133 (2001).
4. Chadwick, D. W. and Otenko, A.: The PERMIS X.509 Role Based Privilege Management Infrastructure: In *Proceedings of the 7th ACM symposium on Access control models and technologies*, Monterey, California, USA, 135–140 (2002).
5. Chadwick, D. W., Otenko, S. and Nguyen, T. A.: Adding Support to XACML for Dynamic Delegation of Authority in Multiple Domains: In *Proceedings of the 10th IFIP Open Conference on Communications and Multimedia Security (CMS 2006)*, Heraklion Crete, Greece, 67–86 (2006).
6. Crampton, J. and Khambhammettu, H.: Delegation in Role-Based Access Control: In *Proceedings of the 11th European Symposium On Research In Computer Security (ESORICS 2006)*, Hamburg, Germany, 174–191 (2006).
7. Crispo, B.: Delegation of Responsibility. Lecture Notes in Computer Science (LNCS) - Security Protocols, **1550**, 626 (1999).
8. Gomi, H., Hatakeyama, M., Hosono, S., and Fujita, S.: A delegation framework for federated identity management: In *Proceedings of the ACM Workshop on Digital Identity Management*, 94–103 (2005).
9. Lorch, M., Adams, D., Kafura, D., Koneni, M., Rathi, A., and Shah, S.: The PRIMA System for Privilege Management, Authorization and Enforcement in Grid Environments: In *Proceedings of the 4th International Workshop on Grid Computing - Grid 2003*, Phoenix, AR, USA (2003).
10. Pham, Q., McCullagh, A., and Dawson, E., Consistency of User Attribute in Federated Systems: In *Proceedings of the 4th International Conference on Trust, Privacy & Security in Digital Business (TrustBus 2007)*, Regensburg, Germany, 165–177 (2007).
11. Schaad, A.: A Framework for Organisational Control Principles. PhD Thesis, The University of York, York, England (2003).
12. Varadharajan, V.: Authentication in mobile distributed environment: In *Proceedings of the 7th IEE European Conference on Mobile and Personal Communications*, 173–188 (1993).
13. Varadharajan, V., Allen, P., and Black, S.: An analysis of the proxy problem in distributed systems: In *Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy*, Oakland, CA, USA, 255–275 (1991).
14. Zhang, L., Ahn, G. L., and Chu, B. T.: A rule-based framework for role based delegation: In *Proceedings of the 6th ACM symposium on Access control models and technologies*, Chantilly, VA, USA, 153–162 (2001.
15. Zhang, X., Oh, S., and Sandhu, R.: PBDM: a flexible delegation model in RBAC: In *Proceedings of the 8th ACM symposium on Access control models and technologies*, Como, Italy, 149–157 (2003).